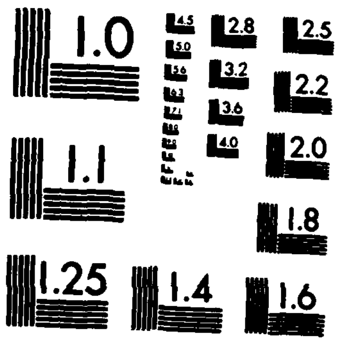
AD-A142 975 COMPUTER PROGRAMS FOR ELECTROMAGNETIC COUPLING BETWEEN A CONDUCTING BODY A..(U) SYRACUSE UNIV NY DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.. S W HSI ET AL. 1/1

UNCLASSIFIED MAR 84 SYRU/DECE/TR-84/4 N00014-76-C-0225 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(8)

SYRU/DECE/TR-84/4

AD-A142 975

COMPUTER PROGRAMS FOR ELECTROMAGNETIC COUPLING BETWEEN A CONDUCTING BODY AND AN APERTURE IN AN INFINITE CONDUCTING PLANE

by

Sandy W. Hsi
Roger F. Harrington

Department of
Electrical and Computer Engineering
Syracuse University
Syracuse, New York 13210

Technical Report No. 24
March 1984

Contract No. N00014-76-C-0225

Prepared for

DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

SYRU/DECE/TR-84/4

COMPUTER PROGRAMS FOR ELECTROMAGNETIC COUPLING BETWEEN A CONDUCTING BODY AND AN APERTURE IN AN INFINITE CONDUCTING PLANE

by

Sandy W. Hsi
Roger F. Harrington

Department of
Electrical and Computer Engineering
Syracuse University
Syracuse, New York 13210

Technical Report No. 24

March 1984

Contract No. N00014-76-C-0225

Prepared for

DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SYRU/DECE/TR-84/4	2. GOVT ACCESSION NO. A142975	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER PROGRAMS FOR ELECTROMAGNETIC COUPLING BETWEEN A CONDUCTING BODY AND AN APERTURE IN AN INFINITE CONDUCTING PLANE		5. TYPE OF REPORT & PERIOD COVERED Technical Report No. 24 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Sandy W. Hsi Roger F. Harrington		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0225
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dept. of Electrical & Computer Engineering Syracuse University Syracuse, New York 13210		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE March 1984 13. NUMBER OF PAGES 46
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aperture and wire
Aperture coupling
Body behind aperture
Computer programs
Coupling through aperture
Electromagnetic Fields
Method of moments

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer programs are given for the analysis of electromagnetic coupling between a straight conducting wire and an aperture of arbitrary shape in a conducting plane. The wire can be either of infinite length or of finite length with arbitrary loads. The excitation is either by a plane wave incident on the conducting plane or by transmission line waves on the wire. An equivalent circuit for the aperture and wire is evaluated.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

I. INTRODUCTION

Computer programs for the analysis of electromagnetic coupling between a thin straight conducting wire and an aperture in an infinite conducting plane are briefly described and listed in this report. The aperture is of arbitrary shape and size. The wire is of finite length (with or without loads) or of infinite length. The excitation is either a plane wave incident from the opposite side of the wire or TEM voltages applied on the wire. The current distributions in the aperture and on the wire are computed. In addition, for the case of TEM voltage excitaton, the power transmitted through the aperture is computed. For the case of plane wave excitation, we evaluate an equivalent circuit of the aperture for the transmission line mode on an infinitely long wire or an arbitrarily loaded wire. It is assumed that the reader is familiar with [1], where the general theory and the method of computation are given.

II. COMPUTER PROGRAM DESCRIPTION

There are two main programs: MAIN1 and MAIN2. MAIN1 is for the case of an infinitely long wire or an arbitrarily loaded wire. MAIN2 is for the case of an unloaded wire of finite length. As shown in Table 1, each main program calls subroutines INDATA, GEOM, CURDIR, IMP, MATRIX, CSMINV, MUL, MUL1, and GAUSS. IMP calls subroutine SICI and function

subprogram P. MATRIX calls subroutines TPARM, SCAINT, VECINT, and TADJ. SCAINT calls subroutine INTGRL, VECINT calls subroutine LININT, and CSMINV calls subroutine DTRMNT. All the main programs, subroutines, and function subprogram are described briefly as follows.

Table 1. A list of main programs, subroutines, and function.

```
             ┌ INDATA
             │ GEOM
             │ CURDIR
             │           ┌ SICI
             │ IMP      <
             │           └ P
MAIN1        │
MAIN2       <│           ┌ TPARM
             │ MATRIX   < SCAINT--INTGRL
             │           │ VECINT--LININT
             │           └ TADJ
             │ CSMINV--DTRMNT
             │ MUL
             │ MUL1
             └ GAUSS
```

(A) Main Programs MAIN1 and MAIN2

For program MAIN1, inputs are defined by DATA statements. An example is

```
DATA NNODE/22/,NEDGE/41/,NFACE/20/,NA/19/
DATA D/0.25/,WL/0.4/,NB/20/,RB/0.001/
DATA HPHI/(-1,0.)/,HTHETA/(0.,0.)/,PHI/90./,THETA/0./
DATA ALAMDA/1./
DATA V1/(1.,0.)/,V2/(0.,0.)/
DATA G1/(0.875,0.)/, G2/(0.667,0.)/
```

Where

NNODE= the total number of nodes of the triangular patching used for the aperture.

NEDGE= the total number of edges of the triangular patching.

NFACE= the total number of faces (patches) of the triangular patching.

NA= the total number of internal edges of the triangular patching.

NB= the total number of current expansion functions on the wire.

D= the distance (in wavelengths) from the wire to the conducting plane.

WL= the length (in wavelengths) on the wire where the evanescent current is assumed to exist.

RB= the radius (in wavelengths) of the wire.

HPHI= the ϕ-component of incident magnetic field (in units of ampere/meter).

HTHETA= the θ-component of incident magnetic field (in units of ampere/meter).

PHI= the incidence angle ϕ_o (in degrees)

THETA= the incidence angle θ_o (in degrees)

ALAMDA= the wavelength (in meters).

V1 ,V2= the amplitudes of TEM voltages (in units of $2Z_0$, Z_0 is the characteristic impedance of the trasmission line formed by the wire and the conducting plane) propagating in the +z and -z directions, respectively.

G1 , G2= the reflection coefficients Γ_1 and Γ_2. (G1= G2= 0. when the wire is matched to Z_0 or is infinitely long.)

Outputs of program MAIN1 are the coefficients of current expansion functions in the aperture and on the wire, the power transmitted through the aperture, and network elements. They are stored in a data file with I/O unit number 21.

The minimum allocations are given by

```
Complex Y(NA,NA),Z(NB,NB),T(NA,NB),TT(NB,NA),YT(NA,NB),
        GZ(NB,NB),DGZ(NB,NB),VI(NB),VM(NA),ZZ(NB,NB),
        YI(NB),YTI(NA),CIA(NA),CIB(NA),YY(NA,NA)
Integer NCONN(NEDGE,3),ITRAK(NEDGE),IMIN(NEDGE)
```

In program MAIN2, inputs are the same as those in MAIN1. However, inputs G1 and G2 are assigned to be zero in the program, and NB is defined as the sum of 2 and the total number of current expansion functions on the wire. WL is defined as the difference of L (the length of the wire) and the length of one subsection, i.e., WL=L/(1+1/NB). Ouputs are the same as those in MAIN1. However, the TEM equivalent circuit is not evaluated. The minimum allocations are given by

```
Complex Y(NA,NA),Z(NB,NB),T(NA,NB),TT(NB,NA),YT(NA,NB-2),
        GZ(NB-2,NB-2),DGZ(NB-2,NB-2),VI(NB-2),ZF(NB-2,NB-2),
        TTF(NB-2,NA),VM(NA),ZZ(NB-2,NB-2),YT(NA),YTI(NA),
        CIA(NA),CIB(NA),YY(NA,NA),TF(NA,NB-2)
Integer NCONN(NEDGE,3),ITRAK(NEDGE),NBOUND(50,4),IMIN(NEDGE)
```

(B) Subroutines INDATA, GEOM, and CURDIR

Subroutine INDATA(DATNOD, NCONN, NNODE, NEDGE) reads two sets of input data from a file with I/O unit number 20. The subroutine then arranges these input data in a numerical order for the triangular patching of the aperture. The first set of data contains node numbers along with their coordinates. This information is stored in an output array DATNOD. The second set contains edge numbers with the node numbers connected by them. This information is stored in an output array NCONN. Note that in the input data file, we enumerate the internal edges fist. That is, if there are N internal edges, the edge numbers of internal edges start at 1, while the boundary edges start at N+1. The input variables are NNODE and NEDGE, which are defined in (A). The minimum allocations are given by

```
Real DATNOD(NNODE,3)
Integer NCONN(NEDGE,3)
```

Subroutine GEOM(NCONN, NBOUND, ITRAK, IMIN, NEDGE) uses the informations stored in the input array NCONN to form triangular patches. Inputs are NCONN and NEDGE. Outputs are array NBOUND storing face (patch) numbers and their associated edge numbers. ITRAK and IMIN are auxiliary arrays needed in the program. Minimum allocations are given by

Integer NCONN(NEDGE, 3), NBOUND(50, 4), ITRAK(NEDGE), IMIN(NEDGE)

Subroutine CURDIR(NCONN, NBOUND, NFACE, NEDGE, IMIN, NSE) arranges the informations in input arrays NCONN and NBOUND and then transfers them into an output data file with I/O unit number 21. Inputs are NCONN, NBOUND, NFACE, and NEDGE, which are defined previously. Input NSE is the total number of boundary edges. IMIN is an auxiliary array. Minimum allocations are given by

Integer NCONN(NEDGE,3),NBOUND(50,4),IMIN(NEDGE)

(C) Suboutines TPARM and TADJ

Subroutine TPARM(N, DATNOD, NCONN, NBOUND, NNODE, NEDGE, EN, NN, XN, ZN, LN) finds the edge numbers, node numbers, the x and the z-coordinates of nodes, and lengths of edges for the triangle whose face number is N. These informations are stored in output arrays EN, NN, XN, ZN, and LN, respectively. Inputs are N, DATNOD, NCONN, NBOUND, NNODE, and NEDGE. Minimum allocations are given by

```
Integer NCONN(NEDGE,3)
Real DATNOD(NNODE,3)
```

Subroutine TADJ(N, NA, EN, NN, NCONN, NEDGE, M, DIR) finds the current reference direction crossing an edge of a triangle, and the expansion function number associated with this edge. They are stored in output variables DIR and M, respectively. DIR=1 if the current direction is away from the triangle. DIR=-1 if the current direction is towards the triangle. DIR=0 if the edge is a boundary edge. The edge is specified by N which is the node number of its free node (node not on the edge). Inputs are N, NA, EN, NN, NCONN, and NEDGE. Minimum allocations are given by

```
Integer NCONN(NEDGE,3)
```

(D) Subroutine MATRIX

Subroutine MATRIX(NA, NB, NNODE, NFACE, DATNOD, NCONN, NBOUND, CIA, CIB, T, TT, Y) computes source vectors $\vec{I}^{ia}$ and $\vec{I}^{ib}$ and matrices [T], [$\hat{T}$], and [Y], which are defined in [1]. These vectors and matrices are stored in output arrays CIA, CIB, T, TT, and Y, respectively. Inputs are NA, NB, NNODE, NEDGE, NFACE, DATNOD, NCONN, and NBOUND.

The minimum allocations are given by

```
Complex CIA(NA),CIB(NA),T(NA,NB),TT(NB,NA),Y(NA,NA)
Real DATNOD(NNODE,3)
Integer NCONN(NEDGE,3)
```

(E) Subroutines IMP, SICI, and Function P

Subroutine IMP(NB, RB, Z) computes the impedance matrix [Z] for an infinitely long wire. Inputs are NB and RB, and output is the [Z] matrix stored in an array Z. Minimum allocations are given by

```
Complex Z(NB,NB)
```

Subroutine SICI(SI,CI,X) computes the sine and cosine integrals

$$SI = \int_0^X \frac{\sin u}{u} du$$

$$CI = \int_{X}^{\infty} \frac{\cos u}{u} du$$

where X is the input, and SI and CI are the outputs. This subroutine is described in [2].

Complex function P(AL, Z, ZL) evaluates the scalar Green's function

$$P = \frac{1}{4\pi\Delta l_n} \int_{\Delta l_n} \frac{e^{-jk\sqrt{(z_m-z')^2+\rho^2}}}{\sqrt{(z_m-z')^2+\rho^2}} dz'$$

The evaluation is described in [3]. Inputs are defined as

AL= $0.5\Delta l_n$ (in wavelengths).

Z= $|z_m-z'|$ (in wavelengths), the distance between the field point and the midpoint of a source element.

ZL= ρ, the transverse coordinate (in wavelengths) of the field point.

The output is P.

<u>(F) Subroutines SCAINT, VECINT, LININT, and INTGRL</u>

Subroutine SCAINT(XS, ZS, X, Z, CPHI, AREA) computes the integral over a source triangle of area A_o,

$$CPHI = \iint_{A_o} \frac{e^{-jkR}}{R} dS$$

where R is the distance between a field point and a source point in the triangle. Inputs are the x and the z coordinates of nodes of the triangle and of the field point. These coordinates are stored in arrays XS and ZS, and variables X and Z, respectively. A_o is stored in the input variable AREA. The Output is CPHI.

Subroutine VECINT(XS, ZS, X, Z, CAXSI, CAETA, AREA) computes

$$CAXSI = \iint_{A_o} \frac{\xi e^{-jkR}}{R} dS$$

$$CAETA = \iint_{A_o} \frac{\eta e^{-jkR}}{R} dS$$

where ξ and η are area coordinates of the source triangle defined in [1]. Inputs are XS, ZS, X, Z, and AREA. Outputs are CAXSI and CAETA.

Subroutine LININT(XS, ZS, X, Z, POTXSI, POTETA, AREA) computes

$$POTXSI = \iint_{A_O} \frac{\xi}{R} dS$$

$$POTETA = \iint_{A_O} \frac{\eta}{R} dS$$

Inputs are XS, ZS, X, Z, and AREA, and outputs are POTXSI and POTETA.

Subroutine INTGRL(XS, ZS, XF, ZF, POT) computes

$$POT = \iint_{A_O} \frac{1}{R} dS$$

Inputs are XS, ZS, XF, ZF, and AREA. XF and ZF are the x and the z coordinates of the field point. The output is POT.

(G) Subroutines CSMINV, DTRMNT, MUL, MUL1, and GAUSS

Subroutine CSMINV(A, NDIM, N) with subroutine DTRMNT inverts a NxN matrix stored in array A. The result is also stored in array A. Inputs are A and NDIM=N.

Subroutine MUL(L, M, N, A, B, C) computes the product of matrices A(L,M) and B(M,N) and stores the result in C(L,N). Inputs are integers L, M, and N and matrices A and

B. The output is C. Minimum allocations are given by

Complex A(L,M), B(M,N), C(L,N)

Subroutine MUL1(L, M, A, B, C) computes the product of matrices A(L, M) and vector B(M) and stores the result in vector C(L). Inputs are integers L and M, matrix A, and Vector B. The output is C. Minimum allocations are given by

Complex A(L,M), B(M), C(L)

Subroutine GAUSS(N, A, B, EPS, ISW) solves a linear system equation $A\underline{X}=B$ by the method of Gaussian elimination. Inputs are the matrix A(N, N), N, and a small constant EPS. The output ISW = 1 if the absolute value of the pivot of column is larger than EPS, and ISW = 0 otherwise. The solution $\underline{X}$ is stored in B as an output. Minimum allocations are given by

Complex A(N,N),B(N)

III. COMPUTER PROGRAM LISTING

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       MAIN1:MAIN PROGRAM FOR THE PROBLEM OF AN ARBITRARILY LOADED
C       OR INFINITELY LONG WIRE BEHIND AN APERTURE OF ARBITRARY SHAPE
C       AND SIZE.
C       THE APERTURE IS IN AN INFINITE CONDUCTING PLANE
C       OF ZERO THICKNESS.
C       THE PLANE IS IN THE X-Z PLANE.
C       THE WIRE IS AT Y=D.
C       EXCITATION IS EITHER A PLANE WAVE FROM THE OPPOSITE SIDE
C       OF THE WIRE OR  TEM VOLTAGES ON THE WIRE.
C       INPUTS:NNODE,NEDGE,NFACE,NA=THE TOTAL NUMBERS OF
C              NODES,EDGES,FACES(PATCHES),INTERNAL EDGES
C              OF TRIANGULAR PATCHING FOR THE PAERTURE.
C              NA IS ALSO THE TOTAL NUMBER OF EXPANSION
C              FUNCTIONS IN THE APERTURE.
C              D:DISTANCE BETWEEN THE WIRE AND THE CONDUCTING PLANE
C              WL:THE RANGE ON THE WIRE WHERE THE EVANESCENT CURRENT
C              IS ASSUMED TO EXIST.
C              NB:TOTAL NUMBER OF EXPANSIONS ON THE WIRE.
C              HPHI,HTHETA,PHI,THETA:THE INCIDENT H-FILEDS AND ANGLES.
C              ALAMDA:WAVELENGTH(IN METERS)(ALL OTHER DIMENSIONS
C       IN UNITS OF ALAMDA)
C              V1,V2:TEM VOLTAGE SOURCE APPLIED ON THE WIRE, REFRRED TO
C              Z=0.(UNITS OF Z0)
C              RE:RADIUS OF THE WIRE
C              G1,G2:RELECTION COEFFICIENTS OF LOADS
C       OUTPUTS: COEFFICIENTS OF CURRENT EXPANSION FUNCTIONS
C              IN THE APERTURE FOR BOTH EXCITATIONS.
C              TOTAL,TEM,EVANESCENT CURRENTS ON THE WIRE.
C              THE TIME-AVE. POWER TRANSMITTED THROUGH TH APERTURE,
C              FOR TEM EXCITATION.
C             AN EQUIVALENT CIRCUIT OF THE APERTURE (SMALL, SYMMETRIC
C             ABOUT Z=0, OR PRODUCING SYMMETRIC TEM OUTWARD TRAVELING
C             CURRENTS) FOR THE TEM MODE ON THE WIRE.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMPLEX Y(19,19),Z(62,62),T(19,62),TT(62,19),HPHI,HTHETA
      COMPLEX YT(19,62),GZ(62,62),DGZ(62,62),VI(62)
      COMPLEX VM(19),ZZ(62,62),YI(19),AI1(2),AI2(2),Z1,Z2,YTI(19)
      COMPLEX E5,V1,V2,JK,CIA(19),CIB(19),G1,G2,PT,YY(19,19)
      COMPLEX C1,C2,C3,C4,C5
      INTEGER NCONN(41,3),ITRAK(41),NBOUND(50,4),IMIN(41)
      COMPLEX VE1,VE2
      REAL DATNOD(22,3)
      EQUIVALENCE(GZ,ZZ)
      COMMON/KKK/AK,PI
      COMMON/JKK/JK
      COMMON/LWIRE/D,WL
      COMMON/FIELD/HPHI,HTHETA,PHI,THETA
      COMMON/WEU/WE,WU
      COMMON/LOAD/G1,G2
      COMMON/VOLT/V1,V2
      COMMON/ZCO/Z0
      DATA NNODE/22/,NEDGE/41/,NFACE/20/,NA/19/
```

```
      DATA D/0.1/,WL/1.5/,NE/62/,RB/0.001/
      DATA HPHI/(-1,0.)/,HTHETA/(0.,0.)/,PHI/90./,THETA/90./
      DATA ALAMDA/1.0/
      DATA V1/(0,0)/,V2/(0.,0.)/
      DATA G1/(0.875,0)/,G2/(0.6667,0)/
      PI=3.14159265
      AK=2.*PI/ALAMDA
      JK=(0.,1.)*AK
      VEL=3.E08
      AOMEGA=AK*VEL
      EPSLON=1.E-09/(36.*PI)
      AMU=4.*PI*1.E-07
      WE=AOMEGA*EPSLON
      WU=AOMEGA*AMU
      NSE=NEDGE-NA
      PHI=PHI*PI/180.
      THETA=THETA*PI/180.
      Z0=60.*ALOG(2.*D/RB)
      OPEN (UNIT=20,FILE='IN.DAT')
      OPEN (UNIT=21,FILE='OUT.DAT')
      CALL INDATA(DATNOD,NCONN,NNODE,NEDGE)
      CALL GEOM(NCONN,NBOUND,ITRAK,IMIN,NEDGE,NNODE)
      CALL CURDIR(NCONN,NBOUND,NFACE,NEDGE,IMIN,NSE)
      WRITE(21,100)NA,NB,RB,WL,D
C
C     FIND IMPEDANCE MATRIX OF WIRE,Z(NB,NB);
C
      CALL IMP(NB,RB,Z)
C
C     FIND COUPLING MATRICES T,TT,ADMITTANCE Y,SOURCE CIA,CIB
C
      CALL MATRIX(NA,NB,NNODE,NEDGE,NFACE,DATNOD,NCONN,
     1 NBOUND,CIA,CIB,T,TT,Y)
      DO 10 M=1,NA
      DO 10 N=1,NA
10    YY(M,N)=Y(M,N)
      CALL CSMINV(Y,NA,NA)
C
C     MATRIX CALCULATIONS:
C
      CALL MUL(NA,NA,NB,Y,T,YT)
      CALL MUL(NB,NA,NB,TT,YT,GZ)
      DO 20 M=1,NB
      DO 20 N=1,NB
20    DGZ(M,N)=GZ(M,N)-Z(M,N)
C
C     K=1:TEM INCIDENT ; K=2: PLANE WAVE INCIDENT
C
      WRITE(21,110)
      DO 30 K=1,2
      DO 32 M=1,NA
32    VM(M)=-CIB(M)
      IF(K .EQ. 1) GO TO 35
      WRITE(21,120)
      DO 34 M=1,NA
34    VM(M)=CIA(M)
35    CALL MUL1(NA,NA,Y,VM,YI)
      CALL MUL1(NB,NA,TT,YI,VI)
      DO 40 M=1,NB
      DO 40 N=1,NB
```

```
 40      ZZ(M,N)=DGZ(M,N)
         CALL GAUSS(NB,ZZ,VI,1E-11,ISW)
         IF(ISW .EQ. 1) GO TO 45
         TYPE 101
 101     FORMAT('   ISW=0,STOP')
         STOP
 45      WRITE(21,130)
         WRITE(21,140) (VI(M),M=1,NB)
         AI1(K)=VI(1)
         AI2(K)=VI(NB)
         CALL MUL1(NA,NB,YT,VI,YTI)
         DO 60 M=1,NA
         VM(M)=YI(M)-YTI(M)
         VM(M)=VM(M)/(120.*PI)
 60      CONTINUE
         WRITE(21,150)
         WRITE(21,140) (VM(M),M=1,NA)
         WRITE(21,220)
C
C        FIND HIGH MODE ELE CURRENT:
C
         N1=(NB-2)/2
         DO  50 M=2,N1+1
         ZN=WL*FLOAT(2*M-NB-1)/FLOAT(2*NB-4)
         M1=NB-M+1
         E5=CEXP(JK*ZN)
         VI(M)=VI(M)-VI(1)*E5
         VI(M1)=VI(M1)-VI(NB)*E5
 50      CONTINUE
         WRITE(21,140) (VI(M),M=2,NB-1)
C
C        FIND TIME-AVE. POWER TRANSM. THRO. APERTURE,PT
C
         IF(K.EQ.2) GO TO 30
         PT=(0.,0.)
         DO 80 M=1,NA
         YTI(M)=(0.,0.)
         DO 90 N=1,NA
 90      YTI(M)=YY(M,N)*VM(N)+YTI(M)
 80      PT=PT+VM(M)*CONJG(YTI(M))
         PT=0.5*REAL(PT)
         WRITE(21,240)PT
 30      CONTINUE
C
C        FIND EQUIVALENT NETWORK:Z1,Z2,VE1,VE2 (NORMALIZED TO Z0 )
C        (I+=1 OR VO=1)
C
         E5=2.+AI1(1)+AI2(1)
         Z1=-(AI1(1)+AI2(1))/E5
         IF(CABS(AI1(1)-AI2(1))/CABS(AI1(1)) .GT. 1.E-02) GO TO 52
         Z1=-2.*AI1(1)*Z0/(1.+AI1(1))
         VE1=-AI1(2)*(2.*Z0+Z1)
         WRITE(21,230) Z1,VE1
         GO TO 17
 52      CONTINUE
         Z2=2.*(1.+AI2(1))/E5/(AI1(1)-AI2(1))
C
C        TO AVOID ANY POSSIBLE SMALL ERROR(SMALL NEGATIVE  RESISTANT)
C        DUE TO THE VERY SMALL ROUNDOFF ERROR IN REAL(I(1)),REAL(I(NB))
C
```

```
55      CONTINUE
        RZ=REAL(Z1)/CABS(Z1)
        IF( RZ .GT. 0.) GO TO 25
        IF( ABS(RZ) .GT. 3.E-03 ) GO TO 27
        Z1=(0.,1.)*AIMAG(Z1)
25      RZ=REAL(Z2)/CABS(Z2)
        IF ( RZ .GT. 0. )GO TO 37
        IF(ABS(RZ) .GT. 3.E-03) GO TO 27
        Z2=(0.,1.)*AIMAG(Z2)
37      CONTINUE
        E5=1.+Z1+Z2
        VE1=-E5*AI1(2)+Z2*AI2(2)
        VE2=E5*AI2(2)-Z2*AI1(2)
        WRITE(21,170) Z1,Z2,VE1,VE2
        GO TO 17
27      WRITE(21,210)
100     FORMAT(//,2X,'NA=',I4,1X,'NB=',I4,1X,'RB=',F6.4,1X,
     1  'WL=',F6.4,1X,'D=',F6.4)
110     FORMAT(/,2X,'TEM WAVE INCIDENT')
120     FORMAT(/,2X,'PLANE WAVE INCIDENT')
130     FORMAT(2X,'ELEC. CURRENT ON WIRE,1ST & LAST ARE TEM AT Z=0-,0+')
140     FORMAT(2X,2E)
150     FORMAT(2X,'MAG. CURRENT DENSITY ON APERTURE/(120.*PI)=')
170     FORMAT(/,2X,'EQUIVALENT NETWORK(NORMALIZED TO Z0): ',/,2X,'Z1=',
     1  2E,2X,'Z2=',2E,/,2X,'VE1=',2E,2X,'VE2=',2E)
210     FORMAT(2X,'*** STOP, NEED LARGER WL & NB ***')
220     FORMAT(2X,'HIGH MODE ELEC. CURRENT ON WIRE')
230     FORMAT(2X,'Z2 IS OPEN',/,' Z1=',2E,2X,'VE1=' 2E)
240     FORMAT(2X,'POWER TRANSM. THRO. APERTURE,PT=',E)
17      CLOSE (UNIT=20,FILE='IN.DAT')
        CLOSE (UNIT=21,FILE='OUT.DAT')
        STOP
        END
```

```
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       MAIN2:MAIN PROGRAM FOR THE PROBLEM CF AN UNLCADED WIRE OF
C       FINITE LENGTH BEHINC AN APERTURE OF ARBITRARY SHAPE AND SIZE.
C       THE APERTURE IS IN AN INFINITE CONDUCTING PLANE
C       OF ZERO THICKNESS.
C       EXCITATICN IS EITHER A PLANE WAVE FRCM THE OPPOSITE SIDE
C       OF THE WIRE OR  TEM VOLTAGES ON THE WIRE.
C       INPUTS:NNODE,NEDGE,NFACE,NA=THE TCTAL NUMBERS CF
C               NODES,EDGES,FACES(PATCHES),INTERNAL ECGES
C               OF TRIANGULAR PATCHING FOR THE PAERTURE.
C               NA IS ALSO THE TOTAL NUMBER OF EXPANSION
C               FUNCTIONS IN THE APERTURE.
C               D:DISTANCE BETWEEN THE WIRE ANC THE CONDUCTING PLANE
C               WL:LENGTH OF WIRE - LENGTH CF CNE SUBSECTION
C               NB:TOTAL NUMBER OF EXPANSIONS CN THE WIRE + 2.
C               HPHI,HTHETA,PHI,THETA:THE INCIDENT H-HILEDS AND ANGLES.
C               ALAMCA:WAVELENGTH (IN METERS)
C               V1,V2:AMPLITUDES OF TEM VCLTAGES (IN UNITS OF Z0)
C               RB:RACIUS OF THE WIRE
C       OUTPUTS: COEFFICIENTS OF CURRENT EXPANSICN FUNCTICNS
C               IN THE APERTURE FCB BOTH EXCITATIONS.
C               CURRENTS ON THE WIRE
C               THE TIME-AVE. POWER TRANSMITTEC THROUGH TH APERTURE,
C               FOR TEM EXCITATICN.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMPLEX Y(19,19),Z(11,11),T(19,11),TT(11,19),HPHI,HTHETA
      COMPLEX YT(19,9),GZ(9,9),CGZ(9,9),VI(9)
      COMPLEX ZP(9,9),TF(19,9),TTF(9,19)
      COMPLEX VM(19),ZZ(9,9),YI(19),AI1(2),AI2(2),Z1,Z2,YTI(19)
      COMPLEX E5,V1,V2,JK,CIA(19),CIB(19),G1,G2,PT,YY(19,19)
      INTEGER NCONN(41,3),ITRAK(41),NBOUNC(50,4),IMIN(41)
      REAL DATNOD(22,3)
      EQUIVALENCE(GZ,ZZ)
      COMMON/KKK/AK,PI
      COMMON/JKK/JK
      COMMON/LWIRE/C,WL
      COMMON/FIELD/HPHI,HTHETA,PHI,THETA
      COMMON/WEU/WE,WU
      COMMON/VOLT/V1,V2
      COMMON/Z00/ZC
      COMMON/LCAD/G1,G2
      CATA NNODE/22/,NEDGE/41/,NFACE/20/,NA/19/
      CATA D/0.25/,WL/0.45/,NB/11/,RB/0.001/
      CATA HPHI/(-1,0.)/,HTHETA/(0.,0.)/,PHI/90./,THETA/90./
      CATA ALAMCA/1.0/
      CATA V1/(1,0.)/,V2/(0.,0.)/
      G1=(0,0)
      G2=(0,0)
      NBF=NB-2
      PI=3.14159265
      AK=2.*PI/ALAMDA
      JK=(0.,1.)*AK
      VEL=3.E08
      AOMEGA=AK*VEL
      EPSLON=1.E-09/(36.*PI)
      AMU=4.*PI*1.E-07
      WE=AOMEGA*EPSLCN
      WU=AOMEGA*AMU
```

```
      NSE=NEDGE-NA
      PHI=PHI*PI/180.
      THETA=THETA*PI/180.
      OPEN (UNIT=20,FILE='IN.DAT')
      OPEN (UNIT=21,FILE='OUT.DAT')
      CALL INDATA(DATNOD,NCONN,NNODE,NEDGE)
      CALL GEOM(NCONN,NBOUND,ITRAK,IMIN,NEDGE,NNODE)
      CALL CURDIR(NCONN,NBOUND,NFACE,NEDGE,IMIN,NSE)
      WRITE(21,100)NA,NB,RB,WL,D
C
C     FIND IMPEDANCE MATRIX OF WIRE,Z(NB,NB);
C
      CALL IMP(NB,RB,Z)
C
C     FIND COUPLING MATRICES T,TT,ADMITTANCE Y,SOURCE CIA,CIB
C
      CALL MATRIX(NA,NB,NNODE,NEDGE,NFACE,DATNOD,NCONN,
     1 NBOUND,CIA,CIB,T,TT,Y)
      DO 10 M=1,NA
      DO 10 N=1,NA
10    YY(M,N)=Y(M,N)
      CALL CSMINV(Y,NA,NA)
C
C     REDUCE Z,T,TT TO ZF,TF,TTF FOR UNLOADED WIRE:
C
      DO 500 M=1,NBF
      DO 510 N=1,NBF
510   ZF(M,N)=Z(M+1,N+1)
      DO 500 N=1,NA
      TTF(M,N)=TT(M+1,N)
500   TF(N,M)=-TTF(M,N)
      NB=NBF
C
C     MATRIX CALCULATIONS:
C
      CALL MUL(NA,NA,NB,Y,TF,YT)
      CALL MUL(NB,NA,NB,TTF,YT,GZ)
      DO 20 M=1,NB
      DO 20 N=1,NB
20    DGZ(M,N)=GZ(M,N)-ZF(M,N)
C
C     K=1:TEM INCIDENT ; K=2: PLANE WAVE INCIDENT
C
      Z0=60.*ALOG(2.*D/RB)
      WRITE(21,110)
      DO 30 K=1,2
      DO 32 M=1,NA
32    VM(M)=-CIB(M)
      IF(K .EQ. 1) GO TO 35
      WRITE(21,120)
      DO 34 M=1,NA
34    VM(M)=CIA(M)
35    CALL MUL1(NA,NA,Y,VM,YI)
      CALL MUL1(NB,NA,TTF,YI,VI)
      DO 40 M=1,NB
      DO 40 N=1,NB
40    ZZ(M,N)=DGZ(M,N)
      CALL GAUSS(NB,ZZ,VI,1E-11,ISW)
      IF(ISW .EQ. 1) GO TO 45
      TYPE 101
```

```
101     FORMAT('   ISW=0,STOP')
        STOP
45      WRITE(21,130)
        WRITE(21,140) (VI(M),M=1,NB)
        CALL MUL1(NA,NB,YT,VI,YTI)
        DO 60 M=1,NA
        VM(M)=YI(M)-YTI(M)
        VM(M)=VM(M)/(120.*PI)
60      CONTINUE
        WRITE(21,150)
        WRITE(21,140) (VM(M),M=1,NA)
C
C       FIND TIME-AVE. POWER TRANSM. THRO. APERTURE,PT
C
        IF(K.EQ.2)GO TO 30
        PT=(0.,0.)
        DO 80 M=1,NA
        YTI(M)=(0.,0.)
        DO 90 N=1,NA
90      YTI(M)=YY(M,N)*VM(N)+YTI(M)
80      PT=PT+VM(M)*CONJG(YTI(M))
        PT=0.5*REAL(PT)
        WRITE(21,240)PT
30      CONTINUE
100     FORMAT(//,2X,'NA=',I4,1X,'NB=',I4,1X,'RB=',F6.4,1X,
     1  'WL=',F6.4,1X,'D=',F6.4)
110     FORMAT(/,2X,'TEM WAVE INCIDENT')
120     FORMAT(/,2X,'PLANE WAVE INCIDENT')
130     FORMAT(2X,'ELEC. CURRENT ON WIRE')
140     FORMAT(2X,2E)
150     FORMAT(2X,'MAG. CURRENT DENSITY ON APERTURE/(120.*PI)=')
240     FORMAT(2X,'POWER TRANSM. THRO. APERTURE,PT=',E)
17      CLOSE (UNIT=20,FILE='IN.DAT')
        CLOSE (UNIT=21,FILE='OUT.DAT')
        STOP
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       INDATA:READS TWO SETS OF INPUT DATA FOR THE         C
C       TRIANGULAR PATCHING OF THE PAERTURE:(1) NODE NOS.   C
C       WITH COORDINATES, STORED IN DATNOD.(2) EDGE NOS.    C
C       AND NODE NOS. CONNECTED BY THEM, SOTRED IN NCONN    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE INDATA(DATNOD,NCONN,NNODE,NEDGE)
        DIMENSION DATNOD(NNODE,3)
        INTEGER NCONN(NEDGE,3)
        DO 10 I=1,NNODE
        READ(20,5)NODE,X,Z
5       FORMAT(I2,2F9.6)
        AN=FLOAT(NODE)
        DATNOD(NODE,1)=AN
        DATNOD(NODE,2)=X
        DATNOD(NODE,3)=Z
10      CONTINUE
        DO 20 I=1,NEDGE
        READ(20,15)NE,NF,NT
15      FORMAT(I3,2I2)
        NCONN(NE,1)=NE
        NCONN(NE,2)=NF
        NCONN(NE,3)=NT
20      CONTINUE
        WRITE(21, 18)
18      FORMAT('1')
        WRITE(21, 19)
19      FORMAT(6X,'THE FOLLOWING IS THE INFORMATION CONCERNING NODES
        1 AND THEIR COORDINATES',/)
        DO 30 I=1,NNODE
        IDUMMY=IFIX(DATNOD(I,1))
        WRITE(21, 21)IDUMMY,DATNOD(I,2),DATNOD(I,3)
21      FORMAT(3X,'NODE NUMBER=',I3,3X,'X-COORDINATE=',F7.4,3X,
        1'Z-COORDINATE=',F7.4)
30      CONTINUE
        WRITE(21, 28)
28      FORMAT('1')
        WRITE(21, 29)
29      FORMAT(10X,'THIS IS THE INFORMATION CONCERNING EDGES , NODES',/)
        DO 40 I=1,NEDGE
        WRITE(21, 31)NCONN(I,1),NCONN(I,2),NCONN(I,3)
31      FORMAT(3X,'EDGE',I3,1X,'IS CONNECTED FROM NODE',1X,I3,1X,
        1'TO NODE',1X,I3)
40      CONTINUE
        RETURN
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       GEOM:USES INFORMATIONS IN NCONN TO FORM TRIANGULAR  C
C       PATCHING.DIMENSION OF NBOUND MUST BE INCREASED FOR  C
C       TOTAL NUMBERS OF FACES > 50.                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE GEOM(NCONN,NBOUND,ITRAK,IMIN,NEDGE)
        INTEGER NCONN(NEDGE,3),NBOUND(50,4),ITRAK(NEDGE)
        INTEGER IMIN(NEDGE)
        COMMON/IF/IFACE
        IFACE=0
        NF1=0
        NF2=0
        DO 100 IJ=1,NEDGE
        ICOUNT=0
```

```
      N1=NCONN(IJ,2)
      N2=NCONN(IJ,3)
      DO 10 I=1,NEDGE
      DO 10 J=2,3
      IF(I.EQ.IJ)GO TO 10
      NA=NCONN(I,J)
      IF(NA.EQ.N1.OR.NA.EQ.N2)GO TO 6
      GO TO 10
6     ICOUNT=ICOUNT+1
      ITRAK(ICOUNT)=I
10    CONTINUE
      MARK1=0
      MARK2=0
75    CONTINUE
      K1=1
      I1=ITRAK(K1)
      DO 15 I=2,ICOUNT
      IF(ITRAK(I).LT.I1)GO TO 12
      GO TO 15
12    I1=ITRAK(I)
      K1=I
15    CONTINUE
      IF(MARK1.EQ.ICOUNT)GO TO 100
      IF(I1.GT.IJ)GO TO 20
      GO TO 31
20    CONTINUE
      N3=NCONN(I1,2)
      N4=NCONN(I1,3)
      IF(N3.EQ.N1.OR.N3.EQ.N2)GO TO 21
      IF(N4.EQ.N1.OR.N4.EQ.N2)GO TO 22
21    NB=N4
      GO TO 23
22    NB=N3
23    CONTINUE
      ICO=0
      DO 25 I=1,NEDGE
      DO 25 J=2,3
      IF(I.EQ.I1)GO TO 25
      NC=NCONN(I,J)
      IF(NC.EQ.NB)GO TO 24
      GO TO 25
24    ICO=ICO+1
      IMIN(ICO)=I
25    CONTINUE
      DO 30 I=1,ICO
      IA=IMIN(I)
      IF(N1.EQ.NCONN(IA,2).OR.N1.EQ.NCONN(IA,3))GO TO 29
      IF(N2.EQ.NCONN(IA,2).OR.N2.EQ.NCONN(IA,3))GO TO 29
      GO TO 30
29    I2=IA
      GO TO 32
30    CONTINUE
31    CONTINUE
      ITRAK(K1)=NEDGE+1
      MARK1=MARK1+1
      GO TO 75
32    IF(I2.LT.IJ)GO TO 74
      IF(IFACE.EQ.0)GO TO 33
      NF1=NBOUND(IFACE,2)
      NF2=NBOUND(IFACE,3)
```

```
 33     IF(IJ.EQ.NF1.AND.I2.EQ.NF2)GO TO 74
        IFACE=IFACE+1
        NBOUND(IFACE,1)=IFACE
        NBOUND(IFACE,2)=IJ
        NBOUND(IFACE,3)=I1
        NBOUND(IFACE,4)=I2
        MARK2=MARK2+1
        MARK1=MARK1+1
        IF(MARK2.EQ.2)GO TO 100
        ITRAK(K1)=NEDGE+1
        GO TO 75
 74     CONTINUE
        ITRAK(K1)=NEDGE+1
        MARK1=MARK1+1
        GO TO 75
 100    CONTINUE
        WRITE(21, 98)
 98     FORMAT('1')
        WRITE(21, 99)
 99     FORMAT(10X,'THIS IS THE INFORMATION CONCERNING FACES, EDGES',/)
        WRITE(21, 101)((NBOUND(I,J),J=1,4),I=1,IFACE)
 101    FORMAT(3X,'FACE',I3,1X,'IS BETWEEN THE EDGES',1X,I3,
        11X,I3,1X,I3)
        DO 120 I=1,IFACE
        DO 120 J=2,4
        ISEDGE=NBOUND(I,J)
        NCOUNT=0
        DO 125 K=1,IFACE
        DO 125 M=2,4
        IF(I.EQ.K.AND.J.EQ.M)GO TO 125
        IF(ISEDGE.EQ.NBOUND(K,M))NCOUNT=NCOUNT+1
 125    CONTINUE
C       IF(NCOUNT.EQ.0)WRITE(21, 150)ISEDGE
 120    CONTINUE
 150    FORMAT(/5X,I3,2X,'IS A BOUNDARY EDGE')
        RETURN
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       TPARM:OBTAINS PARAMETERS OF A TRIANGLE                     C
C       INPUT:N:TRIANGLE NUMBER CONSIDERED                         C
C            DATNOD(NNODE,3):NODE #,X.Z.COOR FOR ALL NODES OF C
C            SYSTEM                                                C
C           NCONN(NNODE,3):EDGE NO.,NODE NO.(FROM-TO)             C
C           NBOUND(50,4):FACE #,EDGE #S FOR ALL TRIANGLES OF  C
C            SYSTEM                                                C
C       OUTPUT:EN(3):EDGE # OF THE TRIANGLE(SAME ORDER AS         C
C             INPUT BY USER)                                       C
C            NN(3):NODE#(ORDERED AS NODE BETWN.EN(1),(2);(2), C
C            (3);(3),(1)                                           C
C             XN(3),ZN(3): X, Z- COOR. OF VERTICES
C             LN(3):LENGTH OF EDGES OPPOSITE NODE NN(1),(2),(3)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE TPARM(N,DATNOD,NCONN,NBOUND,NNODE,NEDGE,
     1  EN,NN,XN,ZN,LN)
        INTEGER EN(3),NN(3)
        INTEGER NCONN(NEDGE,3),NBOUND(50,4)
        REAL LN(3),XN(3),ZN(3),DATNOD(NNODE,3)
        DO 10 M=1,3
  10    EN(M)=NBOUND(N,M+1)
        DO 20 M=1,3
```

```
          I1=EN(M)
          I2=M+1
          IF(I2.GT.3) I2=1
          I2=EN(I2)
          NN(M)=NCONN(I1,2)
          IF(NCONN(I1,2).EQ.NCONN(I2,2)) GO TO 20
          IF(NCONN(I1,2).EQ.NCONN(I2,3))GO TO 20
          NN(M)=NCONN(I1,3)
 20       CONTINUE
          DO 30 M=1,3
          XN(M)=DATNOD(NN(M),2)
 30       ZN(M)=DATNOD(NN(M),3)
          LN(1)=SQRT((XN(2)-XN(3))**2+(ZN(2)-ZN(3))**2)
          LN(2)=SQRT((XN(3)-XN(1))**2+(ZN(3)-ZN(1))**2)
          LN(3)=SQRT((XN(1)-XN(2))**2+(ZN(1)-ZN(2))**2)
          RETURN
          END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C         TADJ:ADJUST EDGE NO. S.T. M OPPOSITE NODE N; ADJUST INDEX M,
C         S.T. M STARTS AT 1 AND EXCLUDES BOUNDARY EDGE; OBTAIN DIR, THE
C         DIRECTION OF THE CURRENT THRO THE EDGE, DIR=1 IF CURRENT AWAY
C         THE TRIANGLE,-1 IF TOWARDS THE TRIANGLE,0 IF IT IS BOUNDARY.
C         INPUT:N-POINTER OF NODE NC CONSIDERED IN A TRIANGLE=1,2,3
C         NSE:# OF SURFACE (BOUNDARY) EDGES
C              EN(3):EDGE # OF THE TRIANGLE (=1,2,...,NEDGE)
C              NN(3):NODE # OF THE TRIANGLE ,SAME ORDER AS IN TPARM
C              ITRAK(NEDGE):AUXILARY VECTOR
C              NCONN(NEDGE,3):NODE
C              NEDGE:# OF EDGES
C         OUTPUT:
C              M:INDEX
C              DIR:DIRECTION OF CURRENT (=1,-1,0)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          SUBROUTINE TADJ(N,NA,EN,NN,NCONN,NEDGE,M,DIR)
          INTEGER NCONN(NEDGE,3),EN(3),NN(3)
          DIR=0.
C
C         ADJUST EDGE #,S.T. M DENOTES THE DEGE # OPPOSITE NODE N
C         IF M IS A BOUNDARY EDGE,JUMP OUT
C
          IF(N .EQ. 1) M=EN(3)
          IF(N .EQ. 2) M=EN(1)
          IF(N .EQ. 3) M=EN(2)
          IF(M.GT.NA) RETURN
C
C         FIND DIR:
C
          N2=N+1
          N3=N+2
          IF(N2 .GT. 3) N2=N2-3
          IF(N3 .GT. 3) N3=N3-3
          DIR=1.0
          IF(NCONN(M,3).EQ.NN(N2) .AND. NCONN(M,2).EQ.NN(N3))DIR=-1.
          RETURN
          END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C         VECINT:THIS,WITH LININT,EVALUTES VECTOR POTENTIAL INTEGRAL
C         OVER A TRIANGLE REGION
C         INPUT:COOR. OF 3 VERTICES OF THE TRIANGLE (XS(3),ZS(3))
C         OBVERATION POINT (X,Z), AREA (OF THE TRIANGLE)
```

```
C       OUTPUT:INTEGRAL IN CAXSI,CAETA
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE VECINT(XS,ZS,X,Z,CAXSI,CAETA,AREA)
        IMPLICIT COMPLEX (C)
        REAL CABS,COS,XS(3),ZS(3)
        COMMON/KKK/AK,PI
        COMMON/VEC/XSI(7),ETA(7)
        CF=CMPLX(0.,0.)
        CG=CMPLX(0.,0.)
        DO 120 I=1,7
        R1=((X-XS(1))-(XS(2)-XS(1))*XSI(I)-(XS(3)-XS(1))*ETA(I))**2
        R2=((Z-ZS(1))-(ZS(2)-ZS(1))*XSI(I)-(ZS(3)-ZS(1))*ETA(I))**2
        R=SQRT(R1+R2)
        CR=CMPLX(0.0,-1.0*AK*R)
        IF(CABS(CR) .LE. 1.0E-06)GO TO 102
        CA=(CEXP(CR)-CMPLX(1.0,0.0))/CMPLX(R,0.)
        CF1=CMPLX(XSI(I),0.)*CA
        CG1=CMPLX(ETA(I),0.)*CA
        GO TO 103
102     CF1=CMPLX(0.,-AK*XSI(I))
        CG1=CMPLX(0.,-AK*ETA(I))
103     IF(I.EQ.1)GO TO 105
        IF(I.EQ.2 .OR. I.EQ.3 .OR. I.EQ.4)GO TO 110
        CF=CF+CF1*CMPLX(0.1259392,0.)
        CG=CG+CG1*CMPLX(0.1259392,0.)
        GO TO 120
105     CF=CF+CF1*CMPLX(0.225,0.)
        CG=CG+CG1*CMPLX(0.225,0.)
        GO TO 120
110     CF=CF+CF1*CMPLX(0.1323942,0.)
        CG=CG+CG1*CMPLX(0.1323942,0.)
120     CONTINUE
        CALL LININT(XS,ZS,X,Z,POTXSI,POTETA,AREA)
        CAXSI=CF*CMPLX(AREA,0.)+CMPLX(POTXSI,0.)
        CAETA=CG*CMPLX(AREA,0.)+CMPLX(POTETA,0.)
150     CONTINUE
        RETURN
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       SCAINT:THIS SUBROUTINE,WITH SUBROUTINE INTGRL,EVALUATE
C       THE SCALR POTENTIAL INTEGRAL OVER A TIANGLE REGION.
C       INPUT:COOR. OF VERTICES OF TRIANGLE XS(3),ZS(3)
C             AREA OF THE TRIANGLE, AREA;OBSERVATION POINT (X,Z)
C            AK,PI IN COMMON KKK
C       OUTPUT:SCALAR POTENTIAL INTEGRAL OVER AREA,STORED IN CPHI
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE SCAINT(XS,ZS,X,Z,CPHI,AREA)
        IMPLICIT COMPLEX (C)
        REAL CABS,COS,XS(3),ZS(3)
        COMMON/KKK/AK,PI
        COMMON/VEC/XSI(7),ETA(7)
        XSI(1)=1.0/3.0
        XSI(2)=0.05971587
        XSI(3)=0.47014206
        XSI(4)=XSI(3)
        XSI(5)=0.79742699
        XSI(6)=0.10128651
        XSI(7)=XSI(6)
        ETA(1)=XSI(1)
        ETA(2)=XSI(3)
```

```
      ETA(3)=XSI(2)
      ETA(4)=XSI(4)
      ETA(5)=XSI(6)
      ETA(6)=XSI(5)
      ETA(7)=XSI(7)
      CF=CMPLX(0.0,0.0)
      DO 120 I=1,7
      R1=((X-XS(1))-(XS(2)-XS(1))*XSI(I)-(XS(3)-XS(1))*ETA(I))**2
      R2=((Z-ZS(1))-(ZS(2)-ZS(1))*XSI(I)-(ZS(3)-ZS(1))*ETA(I))**2
      R=SQRT(R1+R2)
      CR=CMPLX(0.0,-1.0*AK*R)
      IF(CABS(CR).LE. 1.0E-06) GO TO 102
      CF1=(CEXP(CR)-CMPLX(1.0,0.0))/CMPLX(R,0.0)
      GO TO 103
102   CF1=CMPLX(0.0,-AK)
103   IF(I .EQ. 1) GO TO 105
      IF(I.EQ.2 .OR. I.EQ.3 .OR. I.EQ.4)GO TO 110
      CF=CF+CF1*CMPLX(0.1259392,0.0)
      GO TO 120
105   CF=CF+CF1*CMPLX(0.225,0.C)
      GO TO 120
110   CF=CF+CF1*CMPLX(0.1323942,0.0)
120   CONTINUE
      CALL INTGRL(XS,ZS,X,Z,POT)
      CPHI=CF*CMPLX(AREA,0.0)+CMPLX(POT,0.0)
150   CONTINUE
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     LININT:THIS,WITH SUBROUTINE INTGRL,EVAVULATES XSI/R AND
C     ETA/R INTEGRALS OVER A TRIANGLE REGION.THE QUANTITIES DEFINED
C     HERE ARE THE SAME AS THOSE IN THE REFERENCE.
C     AND POT1 FROM INTGRL IS USED.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE LININT(XS,ZS,X,Z,POTXSI,POTETA,AREA)
      REAL XS(3),ZS(3)
      COMMON/POTEN/POT1
      A=(XS(2)-XS(1))**2+(ZS(2)-ZS(1))**2
      B=(XS(3)-XS(1))**2+(ZS(3)-ZS(1))**2
      C=-2.*((X-XS(1))*(XS(2)-XS(1))+(Z-ZS(1))*(ZS(2)-ZS(1)))
      D=-2.*((X-XS(1))*(XS(3)-XS(1))+(Z-ZS(1))*(ZS(3)-ZS(1)))
      E=2.*((XS(2)-XS(1))*(XS(3)-XS(1))+(ZS(2)-ZS(1))*(ZS(3)-ZS(1)))
      F=(X-XS(1))**2+(Z-ZS(1))**2
      A1=(2.*B-C+D-E)*SQRT(B+D+F)+(2.*A+C-D-E)*SQRT(A+C+F)
      A2=4.*(A+B-E)
      A3=A1/A2
      A4=4.*(A+C)*(B+D+F)+4.*F*(B-C-E)-(C+D+E)**2
      A5=8.*SQRT((A+B-E)**3)
      A6=A4/A5
      IF(ABS(A6).LE.1.E-04)GO TO 5
      AL1=2.*SQRT(A+B-E)*SQRT(B+D+F)
      AL2=2.*SQRT(A+B-E)*SQRT(A+C+F)
      AL3=(2.*B-C+D-E)
      AL4=(2.*A+C-D-E)
      AJ1=A3+A6*ALOG(ABS((AL1+AL3)/(AL2-AL4)))
      AJ3=A3+A6*ALOG(ABS((AL2+AL4)/(AL1-AL3)))
      GO TO 6
5     AJ1=A3
      AJ3=A3
6     B1=SQRT(A+C+F)
```

```
      B2=((2.*A+C)*B1-C*SQRT(F))/(4.*A)
      ANUM=ABS(2.*SQRT(A)*B1+2.*A+C)
      DEN=ABS(2.*SQRT(A*F)+C)
      IF(ANUM.LE.1.E-04)GO TO 10
      IF(DEN.LE.1.E-04)GO TO 10
      B3=ABS((2.*SQRT(A)*B1+2.*A+C)/(2.*SQRT(A*F)+C))
      AB3=ALOG(B3)
      AJ4=B2+(4.*A*F-C**2)*AB3/(8.*SQRT(A**3))
      GO TO 11
10    AJ4=B2
11    B4=SQRT(B+D+F)
      B5=((2.*B+D)*B4-D*SQRT(F))/(4.*B)
      ANUM=ABS(2.*SQRT(B)*B4+2.*B+D)
      DEN=ABS(2.*SQRT(B*F)+D)
      IF(ANUM.LE.1.E-04)GO TO 15
      IF(DEN.LE.1.-04)GO TO 15
      B6=ABS((2.*SQRT(B)*B4+2.*B+D)/(2.*SQRT(B*F)+D))
      AB6=ALOG(B6)
      AJ2=B5+(4.*B*F-D**2)*AB6/(8.*SQRT(B**3))
      GO TO 16
15    AJ2=B5
16    CONTINUE
      POT=POT1/(2.*AREA)
      AR1=2.*B*(AJ1-AJ2)-E*(AJ3-AJ4)
      AR2=(2.*AR1-(2.*B*C-E*D)*POT)/(4.*A*B-E**2)
      POTXSI=2.*AREA*AR2
      AR3=4.*A*(AJ3-AJ4)-2.*E*(AJ1-AJ2)-(2.*A*D-E*C)*POT
      POTETA=(2.*AREA*AR3)/(4.*A*B-E**2)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     CURDIR:TO CALCULATE THE NORMAL VECTOR TO THE SURFACE
C     ,BY LISTING THE EDGES ASSOCIATED WITH EACH TRIANGLE
C     IN A SEQUENTIAL MANNER.
C     INPUT:NCONN,NBOUND,NFACE,NEDGE,NSE
C     OUTPUT:NBOUND,IMIN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE CURDIR(NCONN,NBOUND,NFACE,NEDGE,IMIN,NSE)
C
      INTEGER NCONN(NEDGE,3),NBOUND(50,4),IMIN(NEDGE)
      INTEGER IMAX(6)
      IM=1
      IMIN(IM)=1
      M1=0
      DO 999 IJK=1,NFACE
      IK=IJK
      DO 2 I=1,6
      IMAX(I)=0
2     CONTINUE
      IFLAG=0
      DO 4 J=1,IM
      IF(IJK .EQ. IMIN(J))GO TO 1
      IFLAG=1
4     CONTINUE
      IF(IFLAG.EQ.1)GO TO 999
1     J1=0
      I1=0
      L1=0
      I2=NBOUND(IK,2)
      I3=NBOUND(IK,3)
```

```
      I4=NBOUND(IK,4)
      N1=NCONN(I2,2)
      N2=NCONN(I2,3)
      N3=NCONN(I3,2)
      IF(N3.EQ.N1 .OR.N3.EQ.N2)GO TO 5
      GO TO 10
5     N3=NCONN(I3,3)
10    CONTINUE
      I1=1
      ITEMP=I2
11    DO 20 IJ=1,NFACE
      DO 12 J=1,IM
      IF (IJ .EQ.IMIN(J))GC IO 20
12    CONTINUE
      J2=NBOUNC(IJ,2)
      J3=NBOUNC(IJ,3)
      J4=NBOUNC(IJ,4)
      IF(ITEMP.EQ.J2 .OR. ITEMP.EC.J3 .OR. ITEMP.EC.J4)
     1 GO TO 15
      GO TO 20
15    IL=IJ
      GO TO 25
20    CONTINUE
      IF(I1 .EQ.1 .AND. J1.EC.1)GO TO 21
      J1=1
      ITEMP=I3
      GO TO 11
21    IF(I1.EQ.1 .AND. J1.EQ.1 .AND. L1.EC.1) GO TO 23
      L1=1
      ITEMP=I4
      GO TO 11
23    IF(M1.EQ.1) GC TO 999
      M1=M1+1
      IK=IJK
      GO TO 1
25    KN1=NCONN(ITEMP,2)
      KN2=NCONN(ITEMP,3)
      IF(N1.EQ.KN1 .OR. N1.EC.KN2) GO TO 35
      KN3=N1
      GO TC 40
35    IF(N2.EQ.KN1 .OR. N2.EQ.KN2) GO TO 36
      KN3=N2
      GO TO 40
36    KN3=N3
40    J2=NBOUNC(IL,2)
      J3=NBOUNC(IL,3)
      J4=NBOUNC(IL,4)
      IF(J2.EQ.ITEMP) GO TC 59
      IF(NCONN(J2,2).EQ.KN1 .OR.NCONN(J2,2).EQ.KN2)GO TO 57
      KN4=NCONN(J2,2)
      GO TO 68
57    KN4=NCONN(J2,3)
      GO TO 68
59    IF(NCONN(J3,2).EQ.KN1 .OR. NCONN(J3,2).EQ.KN2) GO TO 61
      KN4=NCONN(J3,2)
      GO TO 68
61    KN4=NCONN(J3,3)
68    CONTINUE
      IF(IM.EQ.1) GO TO 115
      IF(ITEMP.EQ.IMAX(6)) GC TC 109
```

```
      IMAX(1)=IMAX(5)
      IMAX(2)=IMAX(4)
      IMAX(3)=IMAX(6)
      GO TO 115
109   IMAX(1)=IMAX(4)
      IMAX(2)=IMAX(6)
      IMAX(3)=IMAX(5)
115   IF(M1.NE.1) GO TO 175
      IF(ITEMP.EQ.NBOUND(IJK,4)) GO TO 165
      IMAX(1)=NBOUND(IJK,4)
      IMAX(2)=NBOUND(IJK,2)
      IMAX(3)=NBOUND(IJK,3)
      M1=0
      GO TO 175
165   IMAX(1)=NBOUND(IJK,3)
      IMAX(2)=NBOUND(IJK,4)
      IMAX(3)=NBOUND(IJK,2)
      M1=0
175   KDUMMY=KN3
      DO 100 I=1,2
      IF(I.EQ.1 .AND. IM.NE.1)GO TO 99
      ID=I+(I-1)*2
      IF(ITEMP.EQ.I2) GO TO 79
      IF(ITEMP.EQ.I3) GO TO 89
      IF(N1.EQ.KN3 .AND. N2.EQ.KN1) GO TO 69
      IF(N1.EQ.KN1 .AND. N2.EQ.KN3) GO TO 69
      IMAX(ID)=I3
      IMAX(ID+2)=I2
      GO TO 99
69    IMAX(ID)=I2
      IMAX(ID+2)=I3
      GO TO 99
79    NN1=NCONN(I3,2)
      NN2=NCONN(I3,3)
      IF(NN1.EQ.KN1 .AND. NN2.EQ.KN3) GO TO 81
      IF(NN1.EQ.KN3 .AND. NN2.EQ.KN1) GO TO 81
      IMAX(ID)=I4
      IMAX(ID+2)=I3
      GO TO 99
81    IMAX(ID)=I3
      IMAX(ID+2)=I4
      GO TO 99
89    IF(N1.EQ.KN3 .AND.N2.EQ.KN1) GO TO 91
      IF(N1.EQ.KN1 .AND. N2.EQ.KN3) GO TO 91
      IMAX(ID)=I4
      IMAX(ID+2)=I2
      GO TO 99
91    IMAX(ID)=I2
      IMAX(ID+2)=I4
99    KN3=KN4
      I2=J2
      I3=J3
      I4=J4
100   CONTINUE
      KN3=KDUMMY
      NA1=NCONN(IMAX(1),2)
      NA2=NCONN(IMAX(1),3)
      NB1=NCONN(IMAX(4),2)
      NB2=NCONN(IMAX(4),3)
      IF(NB1.EQ.NA1 .OR.NB1.EQ.NA2) GO TO 125
```

```
      IF(NB2.EQ.NA1 .OR. NB2.EQ.NA2) GO TO 125
      IDUMMY=IMAX(6)
      IMAX(6)=IMAX(4)
      IMAX(4)=IDUMMY
125   IMAX(2)=ITEMP
      IMAX(5)=ITEMP
      IF(IM.NE.1) GO TO 149
      NBOUND(IK,2)=IMAX(1)
      NBOUND(IK,3)=IMAX(2)
      NBOUND(IK,4)=IMAX(3)
149   NBOUND(IL,2)=IMAX(6)
      NBOUND(IL,3)=IMAX(5)
      NBOUND(IL,4)=IMAX(4)
      IM=IM+1
      IMIN(IM)=IL
      IK=IL
      IF(IM.EQ.NFACE) GO TO 1000
      GO TO 1
999   CONTINUE
1000  CONTINUE
      IF(NSE.EQ.0)GO TO 1001
      WRITE(21, 98)
98    FORMAT('1')
      WRITE(21, 102)
102   FORMAT(10X,'LIST OF EDGES & VERTICES BOUNDING EACH
     1 FACE')
      DO 1999 IJK=1,NFACE
      I2=NBOUND(IJK,2)
      I3=NBOUND(IJK,3)
      I4=NBOUND(IJK,4)
      IF(NCONN(I2,2).EQ.NCONN(I3,2)) GO TO 1005
      IF(NCONN(I2,2).EQ.NCONN(I3,3)) GO TO 1005
      N1=NCONN(I2,3)
      GO TO 1006
1005  N1=NCONN(I2,2)
1006  IF(NCONN(I3,2).EQ.NCONN(I4,2)) GO TO 1010
      IF(NCONN(I3,2).EQ.NCONN(I4,3)) GO TO 1010
      N2=NCONN(I3,3)
      GO TO 1011
1010  N2=NCONN(I3,2)
1011  IF(NCONN(I4,2).EQ.NCONN(I2,2)) GO TO 1015
      IF(NCONN(I4,2).EQ.NCONN(I2,3)) GO TO 1015
      N3=NCONN(I4,3)
      GO TO 1016
1015  N3=NCONN(I4,2)
1016  CONTINUE
      WRITE(21, 1050)IJK,I2,I3,I4,N1,N2,N3
1050  FORMAT(/3X,'FACE',I3,1X,'IS BOUNDED BY EDGES',1X,
     1 I3,1X,I3,1X,I3,2X,'AND VERTICES',1X,I3,1X,I3,1X,I3)
1999  CONTINUE
1001  RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       INTGRL:CALCULATE 1/R INTEGRATION OVER A SOURCE TRIANGLE
C       INPUTS:(XS(3),ZS(3)):COORS. OF NODES OF THE TRIANGLE
C              XF,ZF:COOR. OF FIELD POINT
C       OUTPUTS:POT=POT1:INTEGAL RESULT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE INTGRL(XS,ZS,XF,ZF,POT)
        DIMENSION XS(3),ZS(3)
        COMMON/POTEN/POT1
C
C       FIND THE ENDS OF EACH EADGE(FROM (XM,ZM) TO (XP,ZP)
C       THE EDGE NO IS DEFINED AS 1,2,3 BETWEEN NODES(1,2),(2,3),(3,1) A
C       AND THE DIRECTION IS DEFINED
C       AS FROM NODES 1-2,2-3,3-1,RESPECTIVELY.
C
        POT=0.
        DO 10 I=1,3
        XM=XS(I)
        ZM=ZS(I)
        I1=I+1
        IF(I1 .EQ. 4) I1=1
        XP=XS(I1)
        ZP=ZS(I1)
        XPXF=XP-XF
        ZPZF=ZP-ZF
        XMXF=XM-XF
        ZMZF=ZM-ZF
        XPXM=XP-XM
        ZPZM=ZP-ZM
        AR=ABS(XPXF*ZMZF-XMXF*ZPZF)
        IF(AR .LE. 1.E-12) GO TO 10
        XPXM2=XPXM**2
        ZPZM2=ZPZM**2
        DL=SQRT(XPXM2+ZPZM2)
        RO=AR/DL
        DL2=DL**2
        XPXML=XPXM2/DL2
        ZPZML=ZPZM2/DL2
        DOT=(-XMXF*XPXM-ZMZF*ZPZM)/DL2
        XO=XM+DOT*XPXM
        ZO=ZM+DOT*ZPZM
        RO2=RO**2
        DLP=((XP-XO)*XPXM+(ZP-ZO)*ZPZM)/DL
        DLM=((XM-XO)*XPXM+(ZM-ZO)*ZPZM)/DL
        RP=SQRT(RO2+DLP**2)
        RM=SQRT(RO2+DLM**2)
        SIGN=-((XO-XF)*ZPZM-(ZO-ZF)*XPXM)/(DL*RO)
        RATIO=(RP+DLP)/(RM+DLM)
        POT=POT+SIGN*RO*ALOG(RATIO)
10      CONTINUE
        POT=ABS(POT)
        POT1=POT
        RETURN
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       MATRIX:TO OBTAIN SOURCE VECTORS CIA,CIB,COUPLING MATRICES T,TT.
C       THE WIRE IS INFINITELY LONG OR LOADED. THE APERTURE IS OF
C       ARBITRARY SIZE AND SHAPE,AND IS IN AN INFINITE
C       CONDUCTING PLANE OF ZERO THICKNESS.
C       INPUT:NA,NB:TOTAL # OF EXPANSIONS IN APERTURE AND ON WIRE
C            NNODE,NEDGE,NFACE: # OF NODES,EDGES,FACES OF APERTURE
C            DATNOD(NNODE,3):NODE #, X,Z, COOR. OF ALL NODES
C            NCONN(NEDGE,3):EDGE #,NODE #(FROM,TO)OF ALL EDGES
C            NBOUND(50,4):FACE #,EDGE # OF ALL TRIANGLES
C            ETHETA,EPHI,THETA,PHI:INCIADENT E-FIELD AMPLITUDES & ANGLES
C       OUTPUT:CIA(NA),CIB(NA),T(NA,NB),TT(NB,NA),Y(NA,NA)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE MATRIX(NA,NB,NNODE,NEDGE,NFACE,DATNOD,NCONN,
     1  NBOUND,CIA,CIB,T,TT,Y)
        COMPLEX Y(NA,NA),CIA(NA),T(NA,NB),HPHI,HTHETA
        COMPLEX H1,HX,HZ,FX,FZ,TF,SPOT,CPHI,JK,CAXSI,CAETA
        COMPLEX CIB(NA),TT(NB,NA)
        COMPLEX C1,C2,C3,C4,C5,G1,G2,V1,V2,AIP,AIM,TP,TM,T1(19),TNB(19)
        REAL DATNOD(NNODE,3),XF(3),ZF(3),LF(3),XS(3),ZS(3),LS(3),CT(3,2)
        INTEGER NCONN(NEDGE,3),NBOUND(50,4)
        INTEGER EF(3),NF(3),ES(3),NS(3)
        COMMON/LWIRE/C,WL
        COMMON/FIELD/HPHI,HTHETA,PHI,THETA
        COMMON/KKK/AK,PI
        COMMON/JKK/JK
        COMMON/WEU/WE,WU
        COMMON/LOAD/G1,G2
        COMMON/VOLT/V1,V2
        COMMON/Z00/Z0
C
C       NSB=TOTAL NO. OF  SURFACE EDGE(BOUNDARY); NA=TOTAL NO. OF
C       INTERNAL EDGES
C
        NSB=NEDGE-NA
        DO 60 M=1,NA
        DO 70 N=1,NA
70      Y(M,N)=(0.,0.)
        CIA(M)=(0.,0.)
        CIB(M)=(0.,0.)
        T1(M)=(0.,0.)
        TNB(M)=(0.,0.)
        DO 80 N=1,NB
        TT(N,M)=(0.,0.)
80      T(M,N)=(0.,0.)
60      CONTINUE
        COST=COS(THETA)
        SINT=SIN(THETA)
        COSP=COS(PHI)
        SINP=SIN(PHI)
        HX=HTHETA*COST*COSP-HPHI*SINP
        HZ=-HTHETA*SINT
        C4=G1*G2
        C5=1.-C4
        C1=-G1/C5
        C2=C4/C5
        C3=-G2/C5
        C4=2.*C5
        AIP=((1.-G1)*V1+G1*(1.-G2)*V2)/C4
        AIM=(-G2*(1.-G1)*V1-(1.-G2)*V2)/C4
```

```
      DO 10 IJ=1,NFACE
C
C     FOR EACH FIELD TRIANGLE IJ,OBTAIN TRIANGLE PARAMETERS:
C     EDGE #:EF(3);NODE # :NF(3); X.Z. COOR. OF NODES XF(3),ZF(3);
C     LENGTH OF EDGES:LF(3)
C     OBTAIN:CENTROD:XC,ZC; TESTING COMPONENTS OF EDGES:CT(3,2)
C
      N11=IJ
      CALL TPARM(N11,DATNOD,NCONN,NBOUND,NNODE,NEDGE,
     1 EF,NF,XF,ZF,LF)
      XC=(XF(1)+XF(2)+XF(3))/3.
      ZC=(ZF(1)+ZF(2)+ZF(3))/3.
      CT(1,1)=(XF(2)+XF(3))/2.-XC
      CT(1,2)=(ZF(2)+ZF(3))/2.-ZC
      CT(2,1)=(XF(3)+XF(1))/2.-XC
      CT(2,2)=(ZF(3)+ZF(1))/2.-ZC
      CT(3,1)=(XF(1)+XF(2))/2.-XC
      CT(3,2)=(ZF(1)+ZF(2))/2.-ZC
      DO 40 IJK=1,NFACE
C
C     FOR EACH SOURCE TRIANGLE IJK:
C     OBTAIN:EDGE PARAMETERS:EDGE # ES(3),NODE # NS(3),COOR. OF NODES
C            XS(3),ZS(3); LENGTH OF EDGES LS(3)
C     OBTAIN:AREA(AREA OF TRIANGLE IJK)
C     BY TAKING MAGNITUDE OF VECTOR CROSS PRODUCT OF TWO SIDES
C
      N11=IJK
      CALL TPARM(N11,DATNOD,NCONN,NBOUND
     1 ,NNODE,NEDGE,ES,NS,XS,ZS,LS)
      AREA=(XS(2)-XS(1))*(ZS(3)-ZS(1))-(XS(3)-XS(1))*(ZS(2)-ZS(1))
      AREA=ABS(AREA)/2.
C
C     OBTAIN SCALAR & VECTOR POTENTIAL INTERALS(1/2*AREA EXCLUDED):
C     CPHI , CAXSI,CAETA
C
      CALL SCAINT(XS,ZS,XC,ZC,CPHI,AREA)
      CALL VECINT(XS,ZS,XC,ZC,CAXSI,CAETA,AREA)
      DO 20 IR=1,3
C
C     FOR EACH NODE IR OF THE FIELD TRIANGLE IJ:
C     OBTAIN:FIELD-EDGE-TESTING INDEX:M;DIRJECTION OF CURRENT:DIRJ
C
C
      N11=IR
      CALL TADJ(N11,NA,EF,NF,NCONN,NEDGE,M,DIRJ)
      IF(DIRJ .EQ.0.) GO TO 20
      IF(IJK.NE.1)GO TO 1
C
C     COMPUTE SOURCE VECTOR CIA(M): INCIDENT PLANE WAVE IN REGION A
      RC=XC*SINT*COSP+ZC*COST
      H1=(CT(IR,1)*HX+CT(IR,2)*HZ)*CEXP(JK*RC)
      CIA(M)=CIA(M)+2.*DIRJ*LF(IR)*H1
C
C     COMPUTE COUPLING MATRIX T(M,N), SOURCE CIB(M) FOR TEM IN REGION B
C
      D22=D**2+XC**2
      TL=WL
      TL2=TL/2.
      ZCT=ZC+TL2
      R1=SQRT(ZCT**2+D22)
```

```
      H1=(1.-ZCT/R1)*CEXP(-JK*(R1+TL2))
      D4=LF(IR)*DIRJ*D*CT(IR,1)/(2.*PI*D22)
      TP=D4*2.*CEXP(JK*ZC)
      TM=D4*2.*CEXP(-JK*ZC)
      T1(M)=C1*TM+C2*TP+T1(M)
      TNB(M)=C2*TM+C3*TP+TNB(M)
      T(M,1)=T(M,1)+D4*H1
      ZCT=ZC-TL2
      R1=SQRT(ZCT**2+D22)
      H1=(1.+ZCT/R1)*CEXP(-JK*(R1+TL2))
      T(M,NB)=T(M,NB)+D4*H1
      DO 30 N1=2,NB-1
      ZN=WL*FLOAT(2*N1-NB-1)/FLOAT(2*NB-4)
      RN=SQRT(D22+(ZC-ZN)**2)
      H1=(JK/RN**2+1./RN**3)*CEXP(-JK*RN)*WL/FLOAT(NB-2)
30    T(M,N1)=T(M,N1)+D4*D22*H1
      CIB(M)=AIP*TM+AIM*TP+CIB(M)
C
C     COMPUTE ADMITTANCE MATRIX Y(M,N)
C
1     CONTINUE
      DO 50 IK=1,3
C
C     FOR EACH NODE IK OF THE SOURCE TRIANGLE IJK:
C     OBTAIN:SOURCE-EDGE-INDEX N; CURRENT DIRECTION DIRS
C
      N11=IK
      CALL TADJ(N11,NA,ES,NS,NCONN,NEDGE,N,DIRS)
      IF(DIRS .EQ.0.) GO TO 50
C
C     COMPUTE SCALR POTENTIAL SPOT,VECTOR POTENTIAL IN X. Z. :FX,FZ
C     :THE DOT PRODUCT OF POTENTIALS & TESTING,TF,TS
C
      A2=DIRS*LS(IK)/(4.*PI*AREA*2.)
      FX=A2*((XS(1)-XS(IK))*CPHI+(XS(2)-XS(1))*CAXSI
     1   +(XS(3)-XS(1))*CAETA)
      FZ=A2*((ZS(1)-ZS(IK))*CPHI+(ZS(2)-ZS(1))*CAXSI
     1   +(ZS(3)-ZS(1))*CAETA)
      TF=FX*CT(IR,1)+FZ*CT(IR,2)
      SPOT=-CPHI*A2*2./(0.,1.)/WU
      Y(M,N)=Y(M,N)+4.*DIRJ*LF(IR)*((0.,1.)*WE*TF-SPOT)
50    CONTINUE
20    CONTINUE
40    CONTINUE
10    CONTINUE
C
C     GALERKIN SOLUTION :Y(M,N)=Y(N,M)
C
      DO 90 M=1,NA
      DO 90 N=1,M
      Y(M,N)=(Y(M,N)+Y(N,M))/2.
90    Y(N,M)=Y(M,N)
C
C     OBTAIN TT(M,N)
      DO 100 N=1,NA
      DO 100 M=1,NB
100   TT(M,N)=-T(N,M)
      IF(G1 .EQ. 0. .AND. G2 .EQ. 0.) GO TO 2
C
C     ADD T1,TNB DUE TO LOADS TO MATRIX T:
```

```
C
      DO 110 M=1,NA
      T(M,1)=T(M,1)+T1(M)
110   T(M,NB)=T(M,NB)+TNB(M)
2     CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       IMP:FIND IMPEDANCE MATRIX Z FOR AN INFINITELY-LONG WIRE
C       INPUT:NB:# EXPANSIONS.
C             RB:RADIUS OF WIRE
C       OUTPUT:Z
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE IMP(NB,RB,Z)
        COMPLEX Z(NB,NB),ZMN(5),P,JK,E,E1,E2
        DIMENSION DZ(5)
        COMMON/KKK/AK,PI
        COMMON/JKK/JK
        COMMON/LWIRE/D,WL
        COMMON/WEU/WE,WU
        ALB=WL/FLOAT(2*(NB-2))
C
C       FIND Z(M,N), M,N=(2, NB-1)
C       M=2 , THE 1ST SUBSECTION ON WL, SO M-1 IN MO,N-1 IN NO
C       (MM,MO,MP) ,(NM,NO,NP) ARE THE POINTS ON SUBSECTION M & N
C
        D2=D*2.
        NM=0
        NO=1
        NP=2
        M1=2
        M2=NB-1
        DO 10 M=M1,M2
        MO=2*(M-1)-1
        MP=MO+1
        MM=MO-1
        DZ(1)=ABS(FLOAT(MO-NO)*ALB)
        DZ(2)=ABS(FLOAT(MP-NP)*ALB)
        DZ(3)=ABS(FLOAT(MP-NM)*ALB)
        DZ(4)=ABS(FLOAT(MM-NP)*ALB)
        DZ(5)=ABS(FLOAT(MM-NM)*ALB)
        DO 20 J=1,5
20      ZMN(J)=P(ALB,DZ(J),RB)-P(ALB,DZ(J),D2)
        Z(M,M1)=(0.,1.)*WU*4*ALB**2*ZMN(1)
        Z(M,M1)=Z(M,M1)-(0.,1.)/WE*(ZMN(2)-ZMN(3)-ZMN(4)+ZMN(5))
        Z(M1,M)=Z(M,M1)
        DO 30 J=1,M2-M
        Z(M+J,M1+J)=Z(M,M1)
30      Z(M1+J,M+J)=Z(M,M1)
10      CONTINUE
C
C       FIND Z(M,1)=Z(1,M),Z(M,NB)=Z(NB,M), BY USING PULSE TESTING
C       OF J(M), M=(2,NB-1)
C
        ALS=ALB
        D22=D2**2
        L=(NB-1)/2+1
        TL=WL
        TL2=TL/2.
        E=CEXP(-JK*TL2)
        DO 40 M=2,L
        ZO=WL*FLOAT(2*M-NB-1)/FLOAT(2*NB-4)
        DZZ=(ZO)
        DZ1=ABS(DZZ+TL2)
        DZ2=ABS(DZZ-TL2)
        ZMN(1)=P(ALB,DZ1,RB)-P(ALB,DZ1,D2)
        ZMN(3)=P(ALB,DZ2,RB)-P(ALB,DZ2,D2)
```

```
      ZP=(ZO+ALS)
      ZM=(ZO-ALS)
      ZP1=ABS(ZP+TL2)
      ZP2=ABS(ZP-TL2)
      ZM1=ABS(ZM+TL2)
      ZM2=ABS(ZM-TL2)
      ZMN(2)=P(ALS,ZP1,RB)-P(ALS,ZP1,D2)
     1        -P(ALS,ZM1,RB)+P(ALS,ZM1,D2)
      ZMN(4)=P(ALS,ZP2,RB)-P(ALS,ZP2,D2)
     1        -P(ALS,ZM2,RB)+P(ALS,ZM2,D2)
      E1=AK/WE*2.*ALB*E
      E2=-(0.,1.)/WE*E
      Z(M,1)=E1*ZMN(1)+E2*ZMN(2)
      Z(M,NB)=E1*ZMN(3)-E2*ZMN(4)
      Z(1,M)=Z(M,1)
      Z(NB,M)=Z(M,NB)
      M1=NB-M+1
      Z(M1,1)=Z(M,NB)
      Z(M1,NB)=Z(M,1)
      Z(1,M1)=Z(M1,1)
      Z(NB,M1)=Z(M1,NB)
40    CONTINUE
C
C     FIND Z(1,1)=Z(NB,NB), Z(1,NB)=Z(NB,1)
C
      TL22=TL**2
      U01=TL+SQRT(TL22+RB**2)
      U02=TL+SQRT(TL22+D22)
      CALL SICI(S1,C1,U01*AK)
      CALL SICI(S2,C2,U02*AK)
      E=CEXP(-JK*TL)
      E1=E/WE
      E2=(0.,1.)*E1
      Z(1,1)=-E2*(P(ALS,0.,RB)-P(ALS,0.,D2))
      Z(1,NB)=E2*(P(ALS,TL,RB)-P(ALS,TL,D2))
     1 +1./WE*AK/(2.*PI)*(-C1+C2+(0.,1.)*(S1-S2))
      Z(NB,NB)=Z(1,1)
      Z(NB,1)=Z(1,NB)
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       SICI:COMPUTES SINE AND COSINE INTEGRALS. INPUT IS X;
C       OUTPUTS ARE SI, CI.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE SICI(SI,CI,X)
        Z=ABS(X)
        IF(Z-4.) 1,1,4
1       Y=(4.-Z)*(4.+Z)
        SI=X*(((((1.753141E-9*Y+1.568988E-7)*Y+1.374168E-5)
     1  *Y+6.939889E-4)*Y+1.964882E-2)*Y+4.395509E-1)
        CI=((5.772156E-1+ALOG(Z))/Z-Z*(((((1.386985E-10*Y
     1  +1.584996E-8)*Y+1.725752E-6)*Y+1.185999E-4)*Y+
     2  4.990920E-3)*Y+1.315308E-1))*Z
        RETURN
4       SI=SIN(Z)
        Y=COS(Z)
        Z=4./Z
        U=((((((((4.048069E-3*Z-2.279143E-2)*Z+5.515070E-2)
     1  *Z-7.261642E-2)*Z+4.987716E-2)*Z-3.332519E-3)*Z-
     2  2.314617E-2)*Z-1.134958E-5)*Z+6.250011E-2)*Z+
     3  2.583989E-10
        V=(((((((((-5.108699E-3*Z+2.819179E-2)*Z-6.537283E-2)
     1  *Z+7.902034E-2)*Z-4.400416E-2)*Z-7.945556E-3)*Z+
     2  2.601293E-2)*Z-3.764000E-4)*Z-3.122418E-2)*Z-
     3  6.646441E-7)*Z+2.500000E-1
        CI=Z*(SI*V-Y*U)
        SI=-Z*(SI*U+Y*V)+1.570796
        RETURN
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       P:FIND THE INTEGRAL OF 1/(4*PI*2*AL)*CEXP(-JK*R)/R IN A
C       SUBSECTION OF S*AL
C       Z=DIST. IN Z  BETWEEN FIELD & SOURCE PTS.
C       ZL=DIST. IN ZL(=SQRT(X**2+Y**2))
C       AL=HALF SUBSECTION LENGTH
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        COMPLEX FUNCTION P(AL,Z,ZL)
        COMPLEX PO,JK
        REAL I1,I2,I3,I4,LR,KL
        COMMON/KKK/AK,PI
        COMMON/JKK/JK
        R=SQRT(ZL**2+Z**2)
        PO=CEXP(-JK*R)/(4.*PI)
        IF(R .LT. (10*AL)) GO TO 30
C
C       FOR R >= 10*AL, FIND P=P(A0,A1,A2,A3,A4)
C
        KL=AK*AL
        LR=AL/R
        ZR2=(Z/R)**2
        ZR4=(Z/R)**4
        A00=-1.+3.*ZR2
        A01=3.-30.*ZR2+35.*ZR4
        A0=1.+LR**2/6.*A00+LR**4/40.*A01
        A1=LR/6.*A00+LR**3/40.*A01
        A2=-ZR2/6.-LR**2/40.*(1.-12.*ZR2+15.*ZR4)
        A3=LR/60.*(3.*ZR2-5.*ZR4)
        A4=ZR4/120.
        P=PO/R*(A0+(0.,1.)*KL*A1+KL**2*A2+(0.,1.)*KL**3*A3+KL**4*A4)
        GO TO 40
```

```
C
C       FOR R< 10*AL, FIND P=P(I1,I2,I3,I4):
C
 30     Z1=Z+AL
        Z2=Z-AL
        G1=SQRT(ZL**2+Z1**2)
        G2=SQRT(ZL**2+Z2**2)
C
C       FOR Z<= AL:
C
        I1=ALOG((Z1+G1)*(-Z2+G2)/ZL**2)
        IF(Z .LE. AL) GO TO 50
C
C       FOR Z > AL:
C
        I1=ALOG((Z1+G1)/(Z2+G2))
 50     I2=2.*AL
        I3=Z1/2.*G1-Z2/2.*G2+ZL**2/2.*I1
        I4=2.*AL*ZL**2+(2.*AL**3+6.*AL*Z**2)/3.
        P=P0/(2.*AL)*(I1-JK*(I2-R*I1)
     1  -AK**2/2.*(I3-2.*R*I2+R**2*I1)
     2  +(0.,1.)*AK**3/6.*(I4-3.*R*I3+3.*R**2*I2-R**3*I1))
 40     RETURN
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       CSMINV:INVERTS MATRIX A OF DIMENSION NXN (N=NDIM),AND STORES
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       RESULTS IN A
        SUBROUTINE CSMINV(A,NDIM,N)
        COMPLEX A(NDIM,NDIM),PIVOT(60),AMAX,T,SWAP,DETERM
        COMPLEX U,CMPLX,CONJG
        INTEGER*4 IPIVOT(60),INDEX(60,2)
        REAL TEMP,ALPHA(60),CABS
        COMPLEX CTEMP,CALPHA(60)
        IERR=0
        IF(NDIM .LE. 60) GO TO 5
        IERR=1
        WRITE(3,4) NDIM
4       FORMAT('OCSMINV ERROR, ATTEMPT TO INVERT A MATRIX'
     1  I4,' ON A SIDE,'/' WHEN 60 X 60 IS THE MAXIMUM ALLOWED')
        RETURN
5       CONTINUE
        DETERM=CMPLX(1.0,0.0)
        SUMAXA=0.
        DO 20 J=1,N
        ALPHA(J)=0.0
        CALPHA(J)=(0.0,0.0)
        SUMROW=0.
        DO 10 I=1,N
        CALPHA(J)=CALPHA(J)+A(J,I)*CONJG(A(J,I))
        ALPHA(J)=REAL(CALPHA(J))
10      SUMROW=SUMROW+CABS(A(J,I))
        ALPHA(J)=SQRT(ALPHA(J))
        IF(SUMROW .GT. SUMAXA) SUMAXA=SUMROW
20      IPIVOT(J)=0
        DO 600 I=1,N
        AMAX=CMPLX(0.0,0.0)
        DO 105 J=1,N
        IF(IPIVOT(J) -1) 60,105,60
60      DO 100 K=1,N
```

```
      IF(IPIVOT(K)-1) 80,100,740
 80   CTEMP=AMAX*CONJG(AMAX)-A(J,K)*CONJG(A(J,K))
      TEMP=REAL(CTEMP)
      IF(TEMP) 85,85,100
 85   IROW=J
      ICOLUM=K
      AMAX=A(J,K)
 100  CONTINUE
 105  CONTINUE
      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
      IF ( IROW - ICOLUM) 140,260,140
 140  DETERM= -DETERM
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
 200  A(ICOLUM,L)=SWAP
      SWAP=ALPHA(IROW)
      ALPHA(IROW)=ALPHA(ICOLUM)
      CALPHA(ICOLUM)=SWAP
      ALPHA(ICOLUM)=REAL(CALPHA(ICOLUM))
 260  INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT(I)=A(ICOLUM,ICOLUM)
      U=PIVOT(I)
      ALPHAI=ALPHA(ICOLUM)
      CALL DTRMNT(DETERM,U,ALPHAI)
      CTEMP=PIVOT(I)*CONJG(PIVOT(I))
      TEMP=REAL(CTEMP)
      IF(TEMP) 330,720,330
 330  A(ICOLUM,ICOLUM)=CMPLX(1.0,0.)
      DO 350 L=1,N
      U=PIVOT(I)
 350  A(ICOLUM,L)=A(ICOLUM,L)/U
 380  DO 550 L1=1,N
      IF(L1 - ICOLUM) 400, 550,400
 400  T=A(L1,ICOLUM)
      A(L1,ICOLUM)=CMPLX(0.0,0.0)
      DO 450 L=1,N
      U=A(ICOLUM,L)
 450  A(L1,L)=A(L1,L)-U*T
 550  CONTINUE
 600  CONTINUE
 620  DO 710 I=1,N
      L=N+1-I
      IF(INDEX(L,1) - INDEX(L,2)) 630,710,630
 630  JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
 705  CONTINUE
 710  CONTINUE
      SUMAXI=0.
      DO 910 I=1,N
      SUMROW=0.
      DO 900 J=1,N
 900  SUMROW=SUMROW+CABS(A(I,J))
      IF(SUMROW .GT. SUMAXI) SUMAXI=SUMROW
 910  CONTINUE
```

```
      RETURN
720   WRITE(3,730)
730   FORMAT('0',10('*******')/'OMATRIX IS SINGULAR'/'0'
     1 ,10('*******'))
740   RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     DTRMNT:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE DTRMNT(DETERM,U,A)
      COMPLEX DETERM,U,CMPLX
      REAL CABS
      COMMON /SCAFAC/ISCALE
      DATA ISCALE/0/
      IF(CABS(DETERM) .GT. 1.E-10) GO TO 100
      DETERM=DETERM*1.E10
      ISCALE=ISCALE+1
100   DETERM=DETERM*U/CMPLX(A,0.0)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     GAUSS:SOLVE FOR X:A(N,N)*X(N)=B(N); STORED IN B(N)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE GAUSS(N,A,B,EPS,ISW)
      COMPLEX A(N,N),B(N),C,T
      NM1=N-1
      DO 10 K=1,NM1
      C=(0.0,0.0)
      DO 2 I=K,N
      IF(CABS(A(I,K)).LE.CABS(C)) GO TO 2
      C=A(I,K)
      IO=I
2     CONTINUE
      IF(CABS(C).GE.EPS) GO TO 3
      ISW=0
      RETURN
3     IF(IO.EQ.K) GO TO 6
      DO 4 J=K,N
      T=A(K,J)
      A(K,J)=A(IO,J)
4     A(IO,J)=T
      T=B(K)
      B(K)=B(IO)
      B(IO)=T
6     KP1=K+1
      C=1./C
      B(K)=B(K)*C
      DO 10 J=KP1,N
      A(K,J)=A(K,J)*C
      DO 20 I=KP1,N
20    A(I,J)=A(I,J)-A(I,K)*A(K,J)
10    B(J)=B(J)-A(J,K)*B(K)
      B(N)=B(N)/A(N,N)
      DO 40 K=1,NM1
      I=N-K
      C=(0.0,0.0)
      IP1=I+1
      DO 50 J=IP1,N
50    C=C+A(I,J)*B(J)
40    B(I)=B(I)-C
```

```
      ISW=1
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     MUL:C(L,N)=A(L,M)*B(M,N)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MUL(L,M,N,A,B,C)
      COMPLEX A(L,M),B(M,N),C(L,N),W
      DO 20 I=1,L
      DO 20 K=1,N
      W=(0,0)
      DO 10 J=1,M
10    W=A(I,J)*B(J,K)+W
20    C(I,K)=W
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     MUL1:A(L,M)*B(M)=C(L)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MUL1(L,M,A,B,C)
      COMPLEX A(L,M),B(M),C(L)
      DO 10 I=1,L
      C(I)=(0.,0.)
      DO 10 J=1,M
10    C(I)=A(I,J)*B(J)+C(I)
      RETURN
      END
```

REFERENCES

[1] Sandy W. Hsi, R. F. Harrington, and J. R. Mautz, "Electromagnetic Coupling between a Conducting Body and an Aperture in an Infinite Conducting Plane," Report TR-83-19, Department of Electrical and Computer Engineering, Syracuse University, Sponsored by Office of Naval Research, Contract No. N00014-76-C-0225, Dec. 1983.

[2]. R. F. Harrington and J. R. Mautz,"Reactively Loaded Directive Antennas," Technical Report TR-74-6, Department of Electrical and Computer Engineering, Syracuse University, Sept. 1974.

[3] R. F. Harrington,"Matrix Methods for Field Problems," Proc. IEEE, Vol. 55, No. 2, pp. 136-149, Feb. 1967.

END

FILMED

8 84

DTIC